Oxford Cambridge and RSA

A Level Computer Science

H446/01 Computer Systems

Practice paper – Set 2

Time allowed: 2 hours 30 minutes

Do not use:
a calculator

First name

Last name

Centre number

Candidate number

INSTRUCTIONS

- Use black ink.
- Complete the boxes above with your name, centre number and candidate number.
- Answer **all** the questions.
- Write your answer to each question in the space provided. Additional paper may be used if required but you must clearly show your candidate number, centre number and question number(s).
- Do **not** write in the barcodes.

INFORMATION

- The total mark for this paper is **140**.
- The marks for each question are shown in brackets **[ ]**.
- Quality of extended responses will be assessed in questions marked with an asterisk (*).
- This document consists of **32** pages.

Answer **all** questions.

1 An operating system has to manage a system's resources.

(a) One aspect of this is memory management.

(i) Describe **one** difference between paging and segmentation.

.....
.....

.....
.....

.....
.....

.....
..... **[2]**

(ii) Explain how an operating system may overcome the problem of physical memory being full.

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
..... **[4]**

(b) Another job of an operating system is to deal with interrupts.

(i) State what is meant by the term 'interrupt'.

.....
.....

.....
..... **[1]**

(ii) Describe what happens in the CPU when it receives an interrupt.

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
..... **[5]**

2 Mobile Treasure Hunt is a game played on a mobile phone. The game shows the user's position on a map of their local area. Treasure randomly appears on the map and users must move to the appropriate area to collect the treasure before it disappears.

(a) State the name of a sensor or input device the phone might use when playing Mobile Treasure Hunt and explain why it might be used.

Sensor / Input Device:
.....

Use: .....
.....

.....
..... **[2]**

Below is part of the code from Mobile Treasure Hunt.

```
class Treasure

    private value
    private weight
    private name

    public procedure new(givenName)
        name=givenName
        weight=20
        value=randomInteger(1,20)
    endprocedure

    public procedure
        changeName(givenName)
        name=givenName
    endprocedure

endclass

class TreasureChest inherits

    Treasure private locked

    public procedure new(givenName)
        super.new(givenName)
        locked=false
        value=randomInteger(1,100)
        weight=randomInteger(80,120)
    endprocedure

    public procedure pickLock()
        if getRandomNumber()>0.5 then
            locked=false
        endif
     endprocedur
          e
```

```
endclass
```

Fig. 2.1

(b) Explain what is meant by the term 'encapsulation' with reference to the attribute called `name`.

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
..... **[3]**

(c) Describe what is meant by the term 'inheritance', referring to the code in Fig. 2.1.

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
..... **[3]**

(d) Identify all attributes and methods in the `TreasureChest` class.

Methods: .....
.....

.....
.....

Attributes: .....
.....

.....
..... **[2]**

3 A Little Man Computer (LMC) assembly language program is stored in memory as shown in Fig. 3.1.

0	LDA &7
1	ADD #4
2	OUT
3	HLT
4	6
5	2
6	10
7	15
8	16
9	17

Fig. 3.1

In this variant of LMC the symbols & and # are used to denote different modes of addressing.

(a) Given that the output is 17, state the addressing mode represented by each symbol.

(i) &

.....

[1]

(ii) #

.....

[1]

An assembler is used on the code.

(b) Describe what is meant by the term 'assembler'.

.....

.....

.....

.....

.....

.....

.....

..... **[2]**

(c) Explain how pipelining would help a CPU execute the code in Fig. 3.1 more quickly.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[3]**

4 A bus runs between two cities. There are a number of stops on the bus route labelled StopA, StopB and so on. The timetable for the route is represented as a hash table. For each entry in the hash table the key is the bus stop code and the data attached to it is a (zero indexed) array of the times a bus arrives at the stop. The times are stored as strings.

An extract of the hash table is shown below:

```
times=
{
"StopA":["06:55", "07:25", "07:55", "08:55", "09:55", "11:55", "14:00",
"15:00", "15:30", "16:00"]
"StopB":["06:40", "07:40", "08:40", "09:20", "09:40", "14:00", "15:00",
"16:00", "16:30"]
…
…
}
```

`print(times["StopA"][1])` displays `07:25`

(a) State what the code `print(times["StopB"][4])` displays.

.....
..... **[1]**

(b) Write a function called `timeValue` that given a time stored in a string, returns the equivalent integer (using thousands and hundreds for the hours and tens and units for the minutes). The given string should be assumed to represent the time in the 24-hour clock in the format HH:MM

`timeValue("07:55")` should return 755
`timeValue("15:30")` should return 1530

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....

.....

.....

..... **[3]**

(c) Write code for a function that takes in the name of a stop (`stopName`) and the current time as an integer (`currentTime`) in the format described in part (b) (using thousands and hundreds for the hours and tens and units for the minutes). It should return the time of the next available bus in the string format. If there are no more buses available that day it should return the string "No buses".

Example `nextBus("StopA", 1013)` should return `"11:55"`

```
function nextBus(stopName, currentTime)
```

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

```
endfunction
```

[5]

5 Every bank account has an account number and sort code. The sort code identifies the bank branch (location of the bank) with which the account is held and the account number uniquely identifies the bank account. An extract from a bank's database table is shown in Fig. 5.1.

CustomerID	Forename	Surname	Acc No	Sort Code	Branch Name
145204	Elaine	Murray	14725200	67-34-56	Hull
657875	Jordan	Rogers	62703441	67-45-67	Truro
735951	Monim	Khan	96385547	67-00-11	Cambridge
744078	Tom	Banner	45623929	67-00-11	Cambridge

Fig. 5.1

(a) State why the table in Fig. 5.1 is not in Third Normal Form.

.....
.....

.....
..... **[1]**

(b) Explain how the database could be put into Third Normal Form.

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
..... **[3]**

(c)* The bank needs to ensure the data stored in its database is accurate at all times including when customers deposit or withdraw funds.

Discuss how the bank can ensure the accuracy of its data and the importance of doing so.

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[9]**

6 The XOR operator can be used to encrypt data.

(a) Show the effect of applying XOR on Text and Key, by completing the last row of the table below.

Text	O								C								R							
Value	0	1	0	0	1	1	1	1	0	1	0	0	0	0	1	1	0	1	0	1	0	0	1	0
Key	A								B								C							
Value	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	1
XOR																								

[2]

(b) Show the effect of applying XOR on your answer to part (a) and Key, by completing the first and last rows of the table below.

(a)																								
Key	A								B								C							
Value	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	1
XOR																								

[2]

(c) Explain whether the type of encryption described above is symmetric or asymmetric.

.....

.....

.....

..... **[2]**

(d) Explain why asymmetric encryption is more suited to transactions over the internet than symmetric encryption.

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
..... **[4]**

7 A binary search tree is used to store the names of dog breeds.

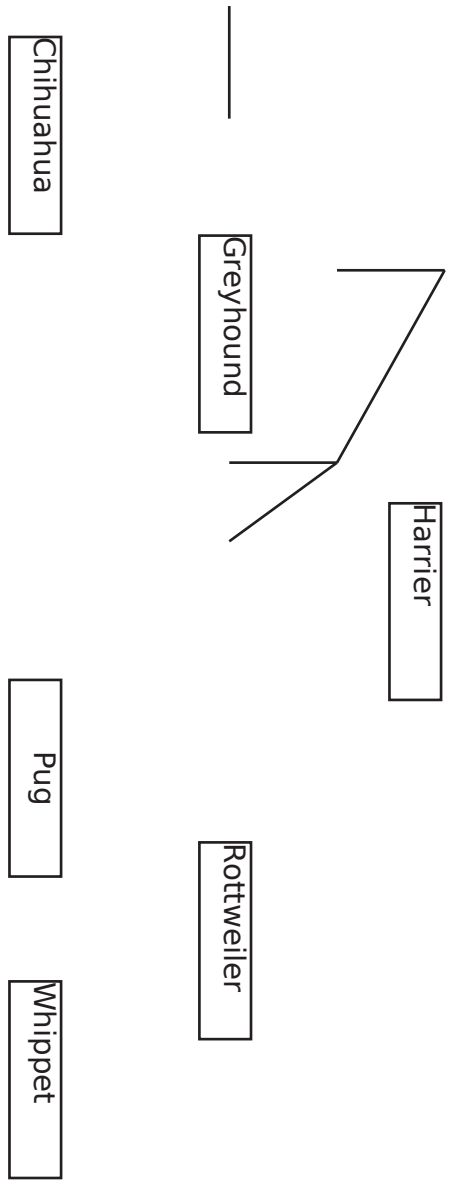


Fig. 7.1

(a) The breeds Doberman and Dalmatian are added to the tree in that order. Add

them to Fig. 7.1.
[2]

(b) Explain how you would determine if the breed Pug is in the binary search tree.

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
..... **[3]**

(c) Explain how you would determine if the breed Spaniel is in the binary search tree.

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
..... **[3]**

(d) The tree is coded using object oriented programming.

Each dog breed is represented by an object of class

Node. The Node class has the methods:
`getLeftNode()` – returns the left hand child node or null if there is no left hand child.
`getRightNode()` – returns the right hand child node or null if there is no right hand child. `getBreed()` – returns the name of the breed stored in that node.

The program allows for a breed name to be entered, and depending on whether the breed is in the tree or not, displays either:

<breed name> `is not in the tree.`

or

<breed name> `is in the tree.`

Complete the program below. Credit will be given for readability of code.

```
name=input("Enter the name of a breed")
breedNode=tree.root() //breedNode is an object of type
Node
                      //representing the root of the tree
```

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....

.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
..... **[6]**

8 A website has the following code.

```
<p>
<form action="checkUser.php">
    Username:<br>
    <input type="text" name="username">
    <br>
    Password:<br>
    <input type="password" name="password">
    <br><br>
    <input type="submit" value="Submit">
</form>

<p id="warning">Unauthorised access to this system will
be prosecuted</p>
```

The page is linked to a style sheet. The message `Unauthorised access to this system will be prosecuted` is red with a monospace font. (Note this is the only text on the page that has this formatting)

(a) Write the segment of CSS code that would appear on the style sheet to make the message appear in the way described.

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
..... **[3]**

(b) Explain the meaning of the HTML line `<input type="text" name="username">`

.....
.....

.....
.....

.....
.....

.....
..... **[2]**

(c)* The line `<form action="checkUser.php">` sends the contents of the form to be processed by the server. This is done by code written in a language called PHP which is designed for server side processing. Conversely JavaScript is traditionally used for client side processing.

Discuss the difference between server and client side processing with respect to webpages. You should refer to the advantages, drawbacks and best uses of both approaches.

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
..... **[9]**

Part of the code on the server can be represented in pseudocode below. In the pseudocode:
`RunSQL(A,B)` runs SQL statement `A` on database `B`. In this case it will always return a single value.
`valueFromForm(controlName)` gets the value entered into the input control with the name `controlName`

```
1  username = valueFromForm("username")
2  password = valueFromForm("password")
3  statement = "SELECT passwordHash FROM users WHERE name = '"
   + username + "'"
4  hashInDB = RunSQL(statement, siteDatabase)
5  if hashInDB == hash(password) then
6     //TO ADD: Generate success webpage
7  else
8     //TO ADD: Store IP address of user in database.
9  endif
```

Fig. 8.1

(d) Explain what the code in Fig. 8.1 does.

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
..... **[5]**

(e) In certain scenarios the user's IP address is logged in a database.

(i) Describe what is meant by an IP Address.

.....

.....

.....

.....

..... **[2]**

(ii) Explain why the programmers have chosen to store the user's IP address.

.....
.....

.....
.....

.....
.....

.....
..... **[2]**

(f) An extract from the database is shown below:

userID	**name**	**passwordHash**
1	admin	0e5a511
2	DenverJ34	f60ccdc
3	TaylorJ22	3a050bc

(i) The username `admin` is entered into the form.

State what the value of `statement` would be after line 03 of the code in Fig. 8 .1 is run.

.....
.....

.....
..... **[1]**

(ii) State what the value of `hashInDB` would be after line 04 of the code in Fig. 8.1 is run.

.....
.....

.....
..... **[1]**

(g) In SQL the character `;` denotes the next statement and the characters `--` denote a comment.

The username `DenverJ34'; DROP TABLE users; --` is entered into the form.

(i) State what the value of `statement` would be after line 03 is run.

.....
.....

.....
.....

.....
.....

.....
..... **[1]**

(ii) Describe what happens when line 04 is run.

.....
.....

.....
.....

.....
.....

.....
..... **[2]**

(iii) State the name of a law the user has broken by entering the username `DenverJ34'; DROP TABLE users; --`

.....
.....

.....
..... **[1]**

9

		AB			
		00	01	11	10
CD	00	1	1	0	1
	01	0	0	0	0
	11	0	0	1	0
	10	1	1	1	0

Fig. 9.1

(a) State the Boolean expression represented by the Karnaugh map in Fig. 9.1, in its smallest form.

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

..... **[4]**

(b) State the simplified versions of the following Boolean expressions:

(i) ¬ ¬A

.....
.....

..... **[1]**

(ii) (¬A ∧ ¬B)

.....
.....

.....
..... **[1]**

(iii) ¬ (¬A ∧ ¬B)

.....
.....

.....
..... **[1]**

10 A NAND gate and its truth table are shown in Fig. 10.1.

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

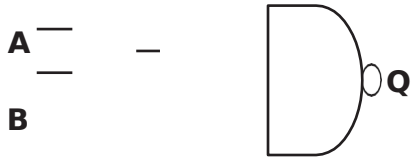


Fig. 10.1

(a) Draw a set of gates equivalent to a NAND gate, but built only of AND, OR and NOT gates.

[2]

The component below is a D-Type, positive edge triggered, flip-flop.

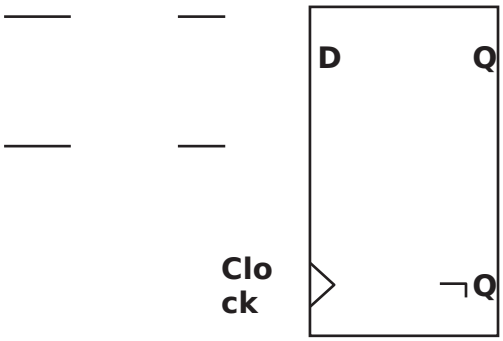


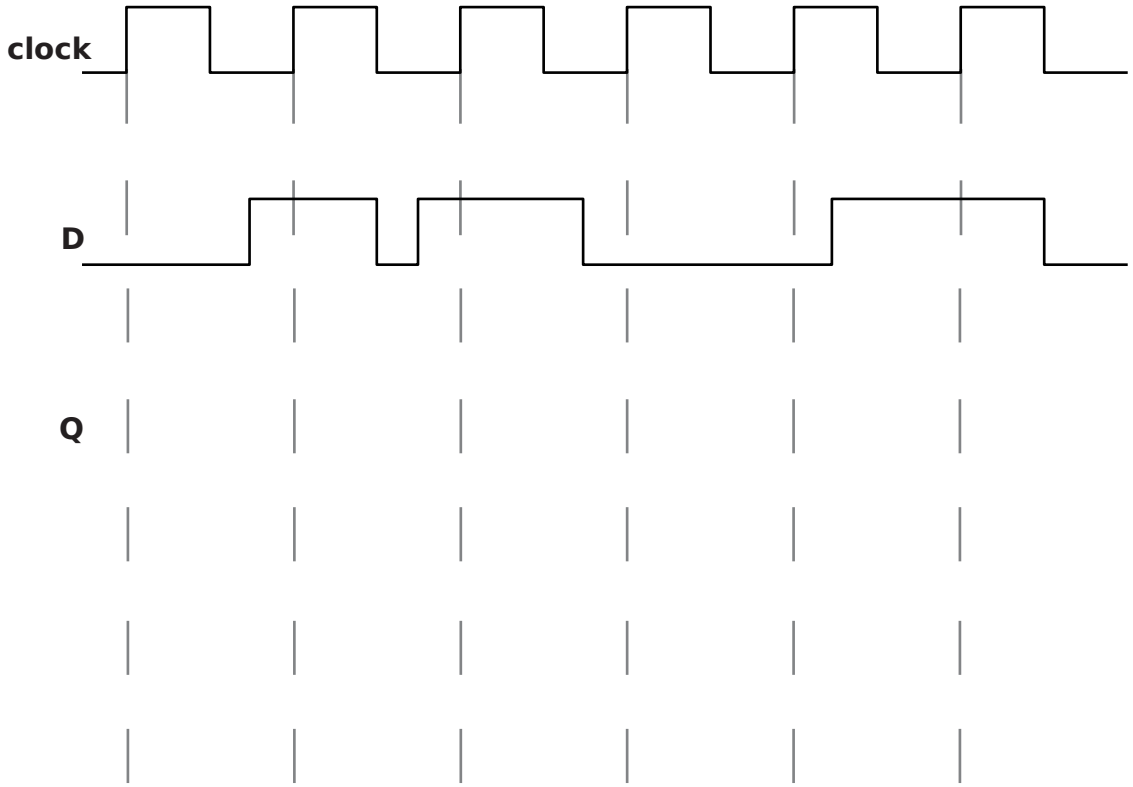
Fig. 10.2

(b) State the purpose of a flip-flop.

.....
.....

.....
..... **[1]**

(c) Draw the output of the flip-flop from Fig. 10.2 on the diagram below.



[3]

11 **(a)** Show a representation of the hexadecimal number AB in:

(i) Binary

.....
.....

.....
.....

.....
.....

.....
..... **[1]**

(ii) Denary

.....
.....

.....
.....

.....
.....

.....
..... **[1]**

(b) Show a representation of denary -119 in 8-bits using:

(i) Sign and Magnitude

.....
.....

.....
.....

.....
.....

.....
..... **[1]**

(ii) Two's Complement

.....
.....

.....
.....

.....
.....

.....
..... **[1]**

(c) A floating point number is represented with a mantissa of 8-bits followed by an exponent of 4-bits, both in two's complement.

00011010 0010

(i) Identify whether or not the number is normalised.

.....
..... **[1]**

(ii) State how you arrived at your answer to part (i).

.....
.....

.....
..... **[1]**

(d) Two floating point numbers are shown below in the same format as used for part (a). Calculate the answer of the second number subtracted from the first. You must show your working and ensure your answer is normalised.

01001100 0011 - 01001010 0010

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
..... **[5]**

12* Some problems require a large amount of computing power that goes well beyond a single CPU.

Discuss the different approaches that can be taken to provide increasingly larger amounts of computing power and the types of problem they are suited to.

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....
.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[12]**

END OF QUESTION PAPER

BLANK

BLANK PAGE

BLANK

BLANK